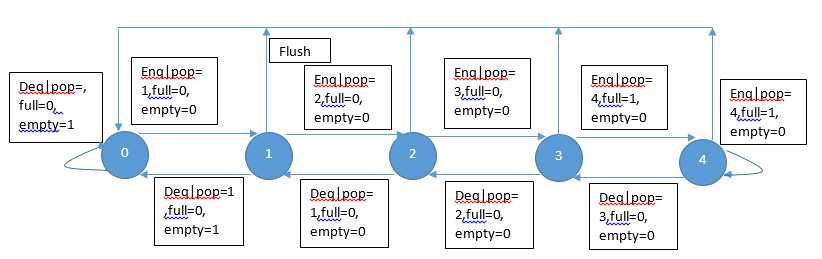
**Goal:** Our goal was to design a First In First Out (FIFO) buffer with parameterizable bit width and capacity. We have designed 5 unit tests to verify the functional correctness of our design.

**Procedure**: We have designed a mixed clock FIFO with capacity 4. The population of the fifo is capped by and needs bits to be represented. On the positive edge of the write clock, if the FIFO is not full and is true, then is stored in the buffer at position, thus increasing the population by 1. Similarly, on the positive edge of the read clock, if the FIFO is not empty and is true, is set to the value at address in the buffer, thus decreasing the population by 1. We also designed 5 test benches to confirm correct functionality (simple behavior, different clock phases and frequencies, enqueue and dequeue happening together, flush, buffer capacity, enqueue on full FIFO, dequeueing from empty FIFO, etc) as shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test Scenario | Clock frequency | Clock phase |
| 1 | 1) Write data\_in in internal buffer till it is full.  2) Read from internal buffer in fifo order | Same | Zero |
| 2 | 1) Read from internal buffer when its empty  2) Write data\_in to internal buffer when its full  3) Read from internal buffer in fifo order | Same | Non-zero |
| 3 | 1) Enqueue and Dequeue signals occur exactly at the same time. | Same | Zero |
| 4 | 1) Ensure FIFO operation with write clock at higher frequency than read clock | Write clock has higher frequency | NA |
| 5 | 1) Increasing the internal buffer and capacity allows larger number of writes and also larger values.  2) Flushing a non empty FIFO, resets population, pointers and the buffer.  3) Ensure FIFO operation with read clock at higher frequency than write clock | Read clock has higher frequency | NA |

**FSM design:** The specifications of a simplifiedMealy machine are shown below and in figure 1. To simplify analysis and representation, we focus on states that represent population. Actually the states also include the pointers, but that just complicates the machine without adding much to our understanding.

**Synthesis:** The following variables are implemented as delay flipflops: population, read\_ptr, write\_ptr, data\_out. Buffer is a memory unit, which has a combinatorial circuit to read and write to appropriate memory locations. The memory locations themselves are flipflops. The ‘if’ statements in the code are synthesized as combinational logic. ‘full’ and ‘empty’ signals, which are generated by assign statements are also combinational. Please refer to figure 2 below.

**Corner case:** Whenboth clocks appear at the same time and both enqueue and dequeue signals are on, then the FIFO behaves as follows: When it is empty, enqueue happens. When it is full, a dequeue happens, but the enqueue is ignored. In other cases, Both the enqueue and dequeue happen. This is a reasonable behavior for this corner case.

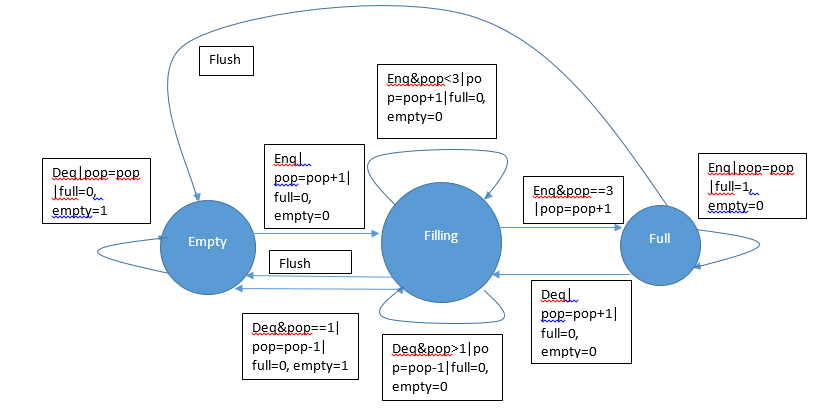


Figure 1: FSM and extended FSM of mealy machine of FIFO

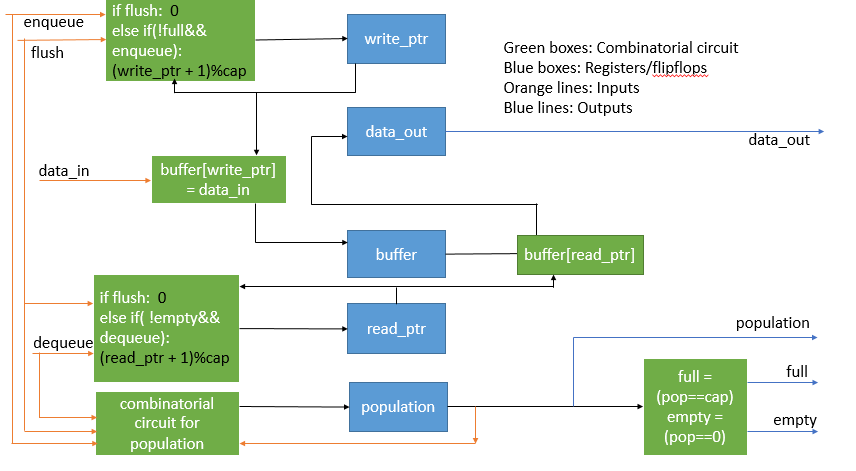


Figure 2: Diagram showing synthesized elements of FIFO